Listing of the Claims

This listing of claims will replace all prior versions, and listings, of the claims:

1. (Previously presented) A method for managing distribution of messages for changing the state of shared data in a computer system having a main memory, a memory management system, a plurality of processors, each processor having an associated cache, and employing a directory-based cache coherency comprising the method of:

grouping the plurality of processors into a plurality of clusters, each cluster having a master processor and at least one slave processor;

tracking copies of shared data sent to processors in the clusters;

receiving an exclusive request from a processor requesting permission to modify a shared copy of the data;

generating invalidate messages requesting that other processors sharing the same data invalidate that data;

sending the invalidate messages to master processors in only those clusters actually containing a processor that has a shared copy of the data in the associated cache; and

distributing the invalidate messages by the master processor to one or more slave processors within each cluster that contains a processor having a shared copy of the data.

2.-8. (Canceled).

9. (Original) A multiprocessor system, comprising:

a main memory configured to store data;

a plurality of processors, each processor coupled to at least one memory cache;

a memory directory controller employing directory-based cache coherence;

at least one input/output device coupled to at least one processor;

a share mask comprising a data register for tracking shared copies of data blocks that are distributed from the main memory to one or more cache locations; and

a PID-SHIFT register which stores configuration settings to determine which one of several shared data invalidation schemes shall be implemented;

wherein when the PID-SHIFT register contains a value of zero, the data bits in the share mask data register correspond to one of the plurality of processors and wherein when the PID-SHIFT register contains a nonzero value, the data bits in the share mask data register correspond to a cluster of processors, each cluster comprising more than one of the plurality of processor.

10. (Original) The system of claim 9 wherein:

if the value in the PID-SHIFT register is zero, the directory controller sets the bit in the share mask corresponding to the processor to which a shared copy of a data block is distributed; and

wherein if the value in the PID-SHIFT register is nonzero, the directory controller sets the bit in the share mask corresponding to the cluster containing a processor to which a shared copy of a data block is distributed.

11.-15. (Canceled).

16. (Previously presented) A multiprocessor system, comprising:

a memory;

multiple computer processor nodes, each with an associated memory cache; and

a memory controller employing a directory-based cache coherency employing shared memory invalidation method, wherein:

the nodes are grouped into clusters, each cluster having a master node and at least one slave node;

the memory controller distributes memory blocks from the memory to the various cache locations at the request of the associated nodes;

upon receiving a request for exclusive ownership of one of the shared memory blocks, the memory controller distributes invalidate messages via direct point to point transmission to a master node in only those clusters containing nodes that share a block of data in the associated cache; and

wherein when the invalidate message is received by a cluster, an invalidate message is distributed by the master node to one or more slave nodes within the cluster.

17.-20. (Canceled).

21. (Previously presented) The method of claim 1 wherein:

distributing the invalidate message within a cluster to one or more other slave processors and waiting for an acknowledgement from said other slave processors; and

a slave processor which does not distribute the invalidate message to any other processor replying to the master processor with an acknowledgement to the processor from which the invalidate message was received.

22. (Previously presented) The method of claim 22 wherein:

upon receiving acknowledgements from all processors within a cluster to which the invalidate message was sent, a slave processor replying with an acknowledgement to the processor from which the invalidate message was received; and

wherein upon receiving an invalidate message, the processor invalidating a local copy of the shared data, if it exists, and wherein upon receiving acknowledgements from all slave processors to which the invalidate messages were sent, the master processor sending an invalidate acknowledgment message to the processor that requested permission to modify the shared copy of the data.

23. (Previously presented) The multiprocessor system of claim 16 further comprising:

a share mask data register with as many bit locations as there are clusters; and

a router lookup table with cross reference information correlating bit locations in the share mask to a master node in each cluster.

24. (Previously presented) The multiprocessor system of claim 23 wherein the memory controller determines to which cluster to send the invalidate message according to bits set in the share mask and sends the invalidate message to the router which then

forwards the invalidate message to the node whose identification corresponds to the cluster number as indicated in the router table.